Romeo System and Status at IU DSC

Andrew J. Younge, Allan Streib, Md Lisul Islam, and Geoffrey C. Fox

Romeo Hardware:

The Romeo system at the IU Digital Science Center consists of 4 compute nodes, complete with Intel CPUs and Nvidia GPUs. Each compute node consists of two sockets of Intel Xeon CPU E5-2670 v3 processors, each of which consist of 12 cores (24 cores total, 48 HT threads) running at 2.3GHz with a TurboBoost frequency of 3.1Ghz. Each node has 128GB of main memory, and is equipped with 4 Nvidia Tesla K80 GPUs. Each of these GPU cards is a dual GPU unit that totals 4990 CUDA cores and 24GB of DDR5 memory.

The 4 Romeo nodes are connected via public and and private Gigabit Ethernet switches, as well as a QDR InfiniBand interconnect for high speed, low latency distributed memory communication. Each node utilizes a Mellanox ConnectX-3 Host Channel Adapter (HCA), with HCAs connected via an Intel Truescale QDR IB switch as part of the Juliet InfiniBand network. This set provides a 1-1 QDR 40Gbs connectivity throughout the nodes, as well as a direct link to the rest of the Juliet system. All nodes are currently installed in the IU Data Center Research pod.

Romeo Software Environment:

The nodes of Romeo are running Red Hat Enterprise Linux 7 currently with kernel 3.10.0-327.22.2. For GPU and InfiniBand usage, CUDA 7.5 was installed with driver v354.39, along with the supported RedHat OFED InfiniBand package, version 3.2.2.0. GCC compiler version 4.8.5 is loaded for general C/C++ codes along with development build essentials, and the Intel 16.0.1 C/C++ and Fortran compilers are also available, along with Boost C++ libraries. For MPI distributed memory communication, both the Intel MPI library as well as the MVAPICH 2.1 GDR libraries are installed. Furthemore, to enable enhanced CUDA enabled Deep Neural Network computations, the cuDNN module has also been installed.

Access to Romeo is granted through the standard FutureSystems account and project allocation mechanisms available on the FutureSystems portal. The SLURM batch scheduling system version 14.11.7 is installed to provide resource scheduling and management of workloads and organized into the “romeo” SLURM partition. User directories are hosted by the existing FutureSystems NFS filesystem mount.

Current Status:

Functionally, the Romeo system is in a fully operational state and is available to FutureSystems users upon request.

The first application deployed on Romeo HOOMD-blue, has been selected as an initial benchmark and acceptance test. This application was selected based on previously lab experience with the software application, its ease of use, and overlap in using many of the libraries originally scheduled for deployment. HOOMD-blue is a Molecular Dynamics Simulation that stresses the machines with MPI, CUDA, and GPUDirect-accelerated codes. Over 50 runs were completed with HOOMD to ensure proper configuration and stability across several days in late May. Specifically, the LJ benchmark at 512k problem size was run across all 96 CPU cores and 32 GPU cores using the above MPI and CUDA libraries. I was able to achieve a max of 1042 million time-steps per second, which is almost a 28% improvement over the 4 node K20+SandyBridge system tested at USC/ISI that was used in previous experiments. Furthermore, we confirmed that the machine was able to meet the most stressful of power and cooling requirements placed on the system at this time without issue.

The first deep learning framework installed on Romeo is the Caffe software, developed by the Berkeley Vision and Learning Center. To facilitate its installation, CUDA, Boost, Intel MKL, and cuDNN libraries were all loaded. Compilation was completed using the GCC compiler and validated through runtime tests. The Caffe installation deep learning framework has been packaged and deployed in a formal mode for all Romeo users to leverage.

Currently underway are efforts to install OpenCV, an open source computer vision and machine learning software library. We also are evaluating Torch, a GPU-forward machine learning library for running efficient ML algorithms on the Nvidia K80 GPUs. Future plans also include expanding deep learning techniques and tools, supported by Md. Lisul Islam, a Ph.D student at Indiana University. Specifically, further usage of the Caffe software will be used across multiple GPUs and in a greater extent, including using PyCaffe. Also, we hope to initiate experimentation with Tensorflow, an open source machine learning library that is capable of constructing deep neural networks.